REMARKS

Claims 1-4, 7-8, 10-12, 14, and 20-23 are pending, with claims 1, 7, 10, 14, 20, 21, 22 and 23 being independent. Claims 1 and 23 have been amended. No new matter has been added. Reconsideration and allowance of the above-referenced application are respectfully requested.

Claim 1 stands rejected under 35 U.S.C. 112, second paragraph. Claim 1 has been amended to expressly state that the node operating to enable the client node to activate remote components on available server nodes is the server node. In view of this amendment, withdrawal of the 112 rejection is respectfully requested.

Claims 1-4 and 23 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Mouko et al. (U.S. Patent No. 6,678,732) in view of Hobbs (U.S. Patent No. 6,523,022). This contention is respectfully traversed.

The claims define systems and techniques whereby a client can generate activation requests to be fulfilled by a server, even if the client lacks information about any specific server that can process such requests. This allows client nodes to create remote components on available server nodes without monitoring the state of the network, where the remote components comprise reusable program building blocks that are combinable

with one or more other components in a distributed network to form an application. (See the present specification at page 2, line 13 to page 3, line 3.) In an embodiment, the invention is implemented within a distributed component object model (DCOM) architecture. (See the present specification at FIG. 2 and page 5, line 17 to page 6, line 20.)

In contrast, Mouko does not describe object-oriented programming or a distributed component object model architecture. Rather, Mouko teaches a dynamic host configuration protocol (DHCP) server, to which only specified DHCP servers send DHCPOFFER messages for DHCPDISCOVER messages sent from DHCP clients. (See Mouko at Abstract.)

The cited portion of Mouko is the background section and prior art figures 14-17, which describe a traditional approach to dynamic allocation of Internet Protocol (IP) addresses to client devices: first, a client broadcasts a DHCPDISCOVER message including IP address lease information; second, the receiving DHCP servers broadcast the requested lease information with a DHCPOFFER message; third, the client collects DHCPOFFER messages and broadcasts a selected IP address lease as a DHCPREQUEST message; finally, a DHCP server decides the IP address for the client and unicasts a DHCPACK message to the client. (See Mouko at col. 1, line8 to col. 2, line 5.) This has nothing whatsoever to do with client activation requests

that active remote program components on servers to form an application, and one skilled in the art would not have believed that the dynamic host configuration protocol for allocating IP addresses to clients has any relation or application to distributed, reusable program building blocks, such as those found in a DCOM architecture.

Hobbs is relied upon for teaching the claimed remote components. However, Hobbs is directed to an apparatus and method for selecting multimedia information residing on a plurality of systems connected to a network, and for linking the multimedia information across the network so that any viewer of a website or other network resource can directly access updated information in real time. (See Hobbs at Abstract.) Hobbs only mentions reusable program building blocks, such as DCOM objects, in passing. Hobbs includes no specific description of how to implement a distributed, reusable program building block architecture.

A prima facie case of obviousness has not been established because there is no suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to combine the reference teachings. The proposed motivation to combine is stated as follows: "because it would have an efficient communications system that using software components that enable to create,

assembly and use of dynamic Java components." However, this proposed motivation cannot find support in the cited references or in the knowledge generally available to one of ordinary skill in the art back at the time of filing of this application on October 29, 1999. In fact, one of skill in the art would have had no reason to believe that DHCP could somehow be combined with DCOM, as is being suggested in the official action.

Moreover, even if the hypothetical combination of Mouko and Hobbs were made, this combination would still not teach or suggest all the claim limitations. The proposed combination would result in the techniques of Mouko being used to allocate IP addresses to clients, and the techniques of Hobbs being used to link information across a network. The techniques of Mouko and Hobbs are most easily understood as being orthogonal to each other. There is nothing in Mouko or Hobbs to suggest that a combination of these references would result in an integration of remote software component activation (such as activation of DCOM objects or dynamic Java software components) into the dynamic host configuration protocol; the process of allocating an IP address to a client is completely separate from and independent of the process of activating a reusable program building block on a server.

Thus, a combination of Mouko and Hobbs still does not result in the features of claim 1: "a client node configured to

process client activation requests; and a server node configured to monitor activation requests from the client node, said server node operating to enable the client node to activate remote components on available server nodes without specific names or capabilities of nodes in the network servicing the requests, wherein said remote components comprise reusable program building blocks that are combinable with one or more other components in a distributed network to form an application." (Emphasis added.) Likewise, such a combination still does not result in the features of claim 23: "client nodes configured to be able to request activation of remote components at run-time without specific names or capabilities of nodes servicing those requests; and server nodes operating to monitor the requests and respond appropriately to service the requests, wherein said remote components comprise reusable program building blocks that are combinable with one or more other components in the distributed component network to form an application." (Emphasis added.)

Dependent claims 2-4 are patentable for at least the above reasons and based on their own merits. Dependent claim 3 specifies that the client node includes enhancements to a network protocol of the client node, and dependent claim 4 specifies that the server node includes enhancements to a network protocol of the server node. Nothing in Mouko suggests

that there are enhancements to a network protocol of either a client node or a server node, as claimed in this context. The cited portion of Mouko (col. 1, lines 32-67) describes the traditional DHCP process; there is no suggesting in the cited portion of Mouko that the network protocols of the client or server have been altered or enhanced in any way from the DHCP standard. Thus, a prima facie showing of unpatentability has not been established for claims 3 or 4.

In addition, independent claim 23 has been amended to further distinguish over the art of record. Claim 23 has been amended to recite, "at least one of said client nodes comprising an augmentation module configured to intercept a remote component activation request from a client-based remote component creation mechanism and configured to broadcast information about the intercepted remote component activation request." The art of record fails to teach or suggest this feature, which is described in detail in the present specification in connection with figures 2, 3 and 5.

Claims 7-8, 10-12, 14, 20 and 22 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Sung et al. (U.S. Patent No. 6,226,684) in view of Hobbs (U.S. Patent No. 6,523,022). This contention is respectfully traversed.

Sung does not describe object-oriented programming or a distributed component object model architecture. Rather, Sung

teaches a method of "reestablishing connections between a particular client and a particular server by multiple routers." (See Sung at col. 13, lines 42-43.) Once a client is assigned to a particular server inside a server bank by a router, this assignment is saved for future use (in a "sticky" Internet Protocol (IP) cache table in the router), and this assignment of a client to a server is multicast to other routers in a collection of routers. Thus, when the client requests additional information from the server bank by sending a request to a router in the collection of routers at a later time, the same server in the server bank handles that information request, which can reduce excessive data caching in the server bank. (See Sung at Abstract; col. 1, line 7 to col. 2, line 32; and col. 5, lines 21-27.)

A prima facie case of obviousness has not been established because no effective suggestion or motivation to combine Hobbs with Sung has been identified, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. The proposed motivation to combine is stated as follows: "because it would have an efficient communications system that using software components that enable to create, assembly and use of dynamic Java components." However, attempting to implement a distributed component object model architecture using Sung's techniques for reestablishing

connections between a particular client and a particular server by using multiple routers, would in fact be a very inefficient approach, and hence, one which would not be selected by one having ordinary skill in the art for this reason.

Sung does not relate to a distributed component model architecture, and there is no indication of a reasonable expectation of success for the suggested combination. Adding multiple routers, each with their own sticky IP cache table, as taught by Sung, to a distributed component object model architecture would add complexity and inefficiency to the architecture without any apparent corresponding benefit. The clients in such a combination would still need to have stored a network address for at least one of the routers before a program object (which corresponds to a remote component provided by one of the servers) could be created.

Moreover, even if the hypothetical combination of Sung and Hobbs were made, this combination would still not teach or suggest all the claim limitations.

With respect to independent claims 7, 14, 20 and 22, the proposed combination would not teach or suggest multicasting a machine-independent activation request to the network as claimed. The multicasting taught by Sung involves multicasting from one router to other routers information indicating the assignment of a client to a server. (See col. 2, lines 14-16;

and col. 11, lines 31-42.) With respect to independent claim 10, the proposed combination would not teach or suggest monitoring, at a server, a specific port to receive a machine-independent client activation request. The server port referred to in Sung is used for all communications from the client. Thus, a prima facie showing of unpatentability has not been established for each of claims 7, 10, 14, 20 and 22.

Dependent claims 8 and 11-12 are patentable for at least the above reasons and based on their own merits. Dependent claim 8 specifies that the capability information includes a list of server IP (Internet Protocol) addresses or UNC (universal naming convention) names of servers that have the ability to service a request for a specific CLSID (Class Identifier). The cited portion of Sung (figures 6-7) shows tables that are used by a router to implement sticky IP in the routers and to redirect client communications to servers in a data center. Sung neither teaches nor suggests a client receiving capability information and/or CLSIDs, which are globally unique identifiers used to refer to particular classes of objects in a distributed component object model architecture, from servers available to service a previously multicast activation request.

Dependent claim 11 specifies that monitoring the specific port at the server includes monitoring a port that is tied to a

multicast IP address. In contrast, the port that is tied to a multicast address in Sung is monitored at the routers and not at the servers. For all of these reasons, claims 1-4, 7-8, 10-12, 14, 20, 22 and 23 should be patentable over the art of record.

Claim 21 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Chung et al. (U.S. Patent No. 6,470,389) in view of Hobbs (U.S. Patent No. 6,523,022). This contention is respectfully traversed.

Chung teaches methods and apparatus for hosting a network service on a cluster of servers, where a common cluster address is used as a secondary address for each server in the cluster, and client requests directed to the cluster address are assigned to the servers using a client-address hashing technique to ensure that only one of the servers of the cluster responds to a given client request. (See Chung at Abstract.) Nothing in Chung suggests returning capability information of a server in response to receipt of a machine-independent client activation request obtained from monitoring a specific port that is tied to a multicast IP address at the server. (Emphasis added.)

A prima facie case of obviousness has not been established because (1) no effective suggestion or motivation to combine Hobbs with Chung has been identified, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, and (2) a combination of Hobbs and

Chung would still not teach or suggest all the claim limitations. The proposed combination would result in the techniques of Chung being used to select servers in a cluster, and the techniques of Hobbs being used to link information across a network. There is nothing in Chung or Hobbs to suggest that a combination of these references would result in an integration of remote software component activation (such as activation of DCOM objects or dynamic Java software components) into the process of selecting a server in a cluster. Thus, claim 21 should be patentable.

For all of these reasons, it is respectfully suggested that the rejection does not meet the patent office's burden of providing a prima facie showing of unpatentability. Thus, all pending claims, 1-4, 7-8, 10-12, 14, and 20-23, should be in condition for allowance.

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific issue or comment does not signify agreement with or concession of that issue or comment. Because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any

claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

It is respectfully suggested for all of these reasons, that the current rejection is totally overcome; that none of the cited art teaches or suggests the features which are now claimed, and therefore that all of these claims should be in condition for allowance. A formal notice of allowance is thus respectfully requested.

No fees are believed due with this response. Please apply any necessary charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 12/23/04

William E. Hunter
Reg. No. 32,030
Attorney for Intel Corporation

Fish & Richardson P.C.
PTO Customer Number: 20985
12390 El Camino Real
San Diego, CA 92130
Telephone: (858) 678-5070
Facsimile: (858) 678-5099
10446853.doc